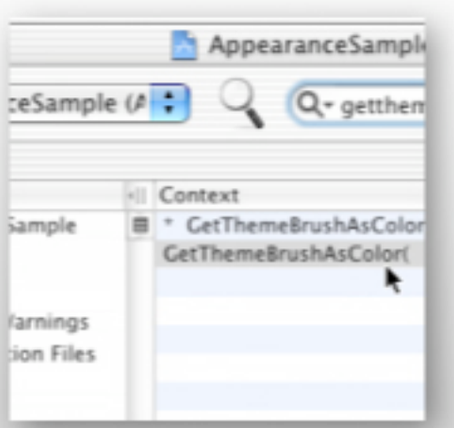
Apple Tools for UNIX Developers

Session 301



Apple Tools for UNIX Developers

Stan Shebs
Senior Engineer, Core Tools

Introduction

- Xcode is a full tool suite
- Includes traditional command-line tools
- Speed ports and ease maintenance
- A superior home for Unix applications
  - "Things Just Work"

What You'll learn

- What tools are delivered with Mac OS X
- Unique features of Mac command-line tools
- Porting issues
- Mixing with Xcode
- Documentation and resources

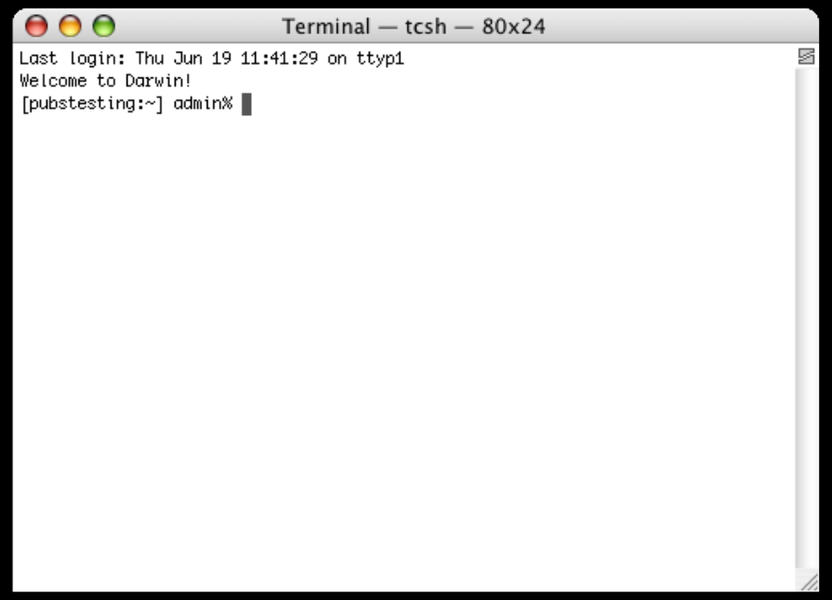
Xcode

Xcode Includes Command-line Tools!

- Familiarity: 30 years of productivity on Unix
- Power: tools are composable
  - diff old.cc new.cc | grep fo[po]_fn
- Leverage: thousands of tools across the industry

Terminal Program

- Launch Terminal
- Uses multiple windows
- Each window a separate shell
- Highly configurable
- Panther
  - Includes server menu



The Catechism

```
% emacs foo.c
% cat foo.c
main() { printf("hello\n"); }
% cc -g foo.c
% ./a.out
hello
% gdb a.out
(gdb) break main
Breakpoint 1 at 0x1eac: file foo.c, line 1.
(gdb) run
…  1 main() {printf("hello\n"); }
(gdb) step
hello
…
```

Shells

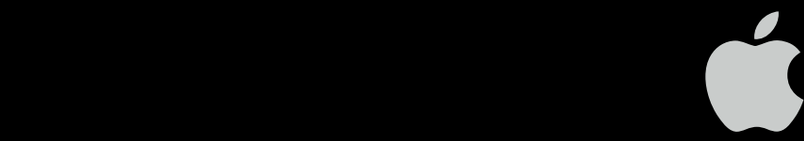
- The shell interprets commands
- Different shells have different scripting languages
  - tcsh
  - zsh
  - bash

Editor Options

- emacs, the one true editor
  - Highly customizable
  - Jaguar standard version is 21.1.1, Panther is 21.2
  - X11, Carbon versions available
- vi
  - “Because we value diversity”
  - Simple and easy to learn
  - Panther uses vim
- pico, sed, ed, cat, *etc*

Language Options

- Supplied
  - C (C90, C99)
  - C++
  - Objective-C, Objective-C++
  - Java (1.3.1, 1.4.1)
  - perl, tcl, PHP, Python, Ruby
- Third parties
  - G77 (Fortran), GNAT (Ada), Lisp



Build Tool Options

- gnumake
  - The standard
- bsdmake
  - For compatibility only
- pbbuild
  - Used by Xcode projects

Other Useful Tools

- Collection most similar to FreeBSD
  - CVS, RCS, SCCS for source code control
  - troff, nroff, eqn, pic, etc., text tools
  - lex, yacc, flex, bison
  - ranlib, ar, as, otool, autoconf
- >400 programs in /bin and /usr/bin
- Ports Manager helps you add to your collection

DarwinPorts

- Similar to Fink and FreeBSD Ports packages
  - Fink and DarwinPorts can co-exist on system
  - Over 400 ports to date
- CLI and GUI interfaces
- Port:
  - fetch
  - extract
  - patch
  - configure
  - build
  - install
  - package

Demo:
DarwinPorts

Sean Fagan
Engineer

APIs

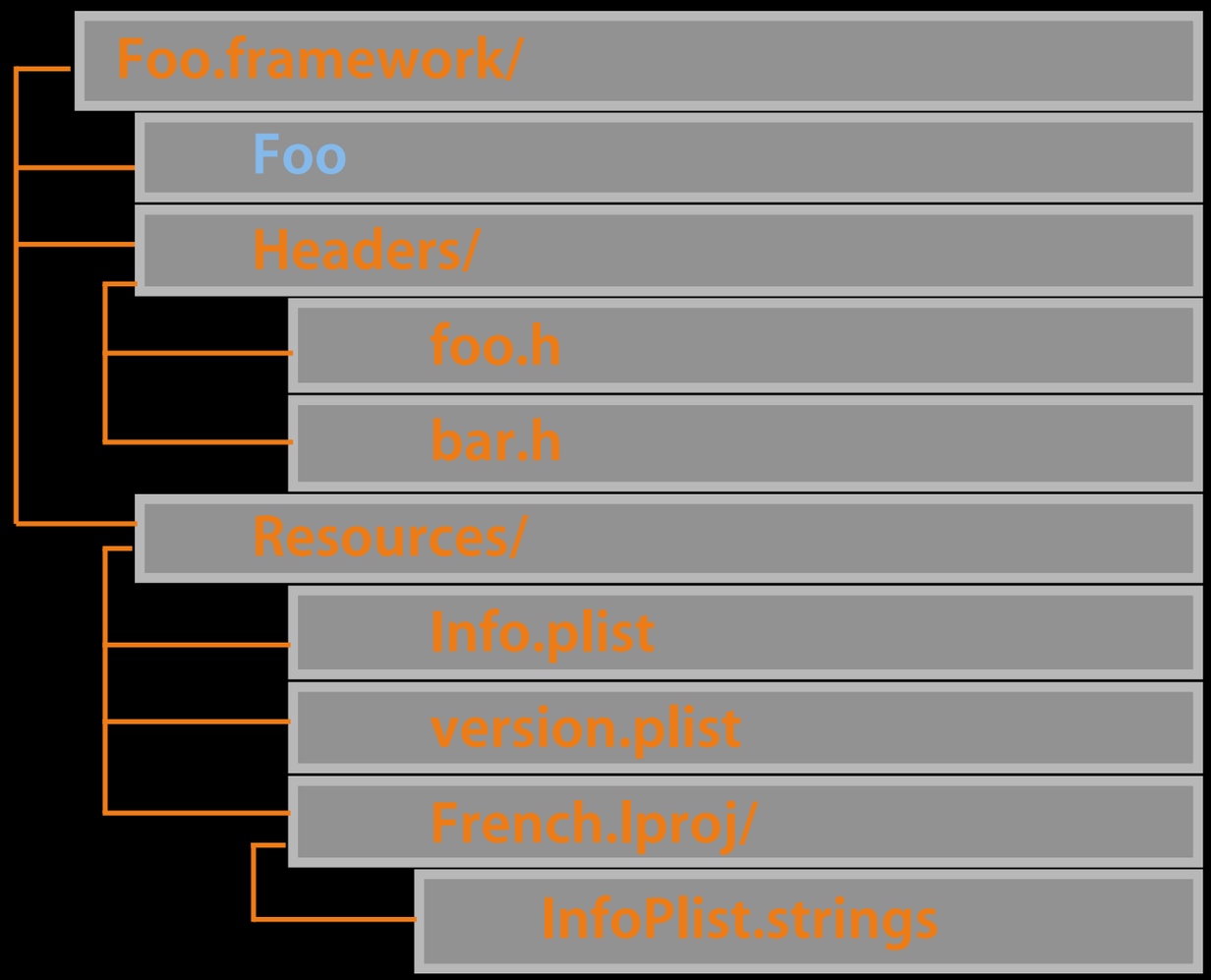
- POSIX
- OpenGL and GLUT
- Carbon and Cocoa for GUI
- IO Kit for drivers and other kernel extensions
- X11 (R6.6) is integrated with Panther!
- IPv6 via CFNetwork

New In Panther

- poll()
- dlopen() and friends
- Asynchronous I/O
- libxml2
- Wide char library
- libiconv
- ICU (International Components for Unicode)
- FreeBSD 5.0 base

Frameworks

- A framework consists of a shared library (dylib), headers, and resources



Using Frameworks

- #include <Foo/foo.h> to compile
- -framework Foo to link
- Most in /System/Library/Frameworks
- -F *path* add place to find frameworks
  - Affects both compiler and linker
- /usr/include still has “usual” headers

Basic Compiler Options

- The Mac OS X compiler *is* GCC
- The usual GCC options work
- Many additional options and language extensions
  - -faltivec enables AltiVec types and built-in functions
    - More powerful than -maltivec in FSF GCC
  - -mdynamic-no-pic for apps
    - Makes nonrelocatable but can still access relocatable dylibs
    - More efficient than default code gen

Building Libraries

- ar + ranlib to create statically linked libraries (libfoo.a)
- cc -dynamiclib for dynamically linked libraries (dylibs)
- GNU libtool (aka glibtool) *usually* knows how to build
  - May need to update libtool bits in source distro

Versions

- Lower layers of system called “Darwin”
  - Open source, no frameworks, no GUI
- sw_vers reports Mac OS X version—10.2.6, etc.
- uname -a reports Darwin version—6.6, etc.
- Mapping
  - Mac OS X v10.0 == Darwin 1.3.1
  - Mac OS X v10.1 == 1.4
  - Mac OS X v10.1.[123] == 5.[123]
  - Mac OS X v10.2.[123456] == 6.[123456]
  - Panther == 7

Basic Porting

- GNU configure/make usually “just works”
- Maybe update config.guess and config.sub
- Do not rely on predefined macros
  - Test for features
  - Add -DMACOSX in makefile

Interface Porting

- Curses (ncurses) “just works”
- Aqua-fied tcl/tk and QtMac are now available
- X11 “just works”
  - X11 apps autolaunch X11 server
- OpenGL and GLUT are standard
  - Must translate

#include <GL/gl.h> ⟶ #include <OpenGL/gl.h>

Common Problems

- 3.1 compiler uses cpp-precomp by default
  - Rarely useful for Unix ports
  - Use -no-cpp-precomp to suppress
- cpp-precomp *not* used by 3.3
- Missing declarations
- Missing functions
  - Reimplement or #ifdef sources
- Duplicate definitions
  - Use -multiply_defined suppress

Less Common Problems

- Some pthread functionality missing
- Namespace conflicts
  - Use -twolevel_namespace
- Run ranlib after ar
- ncurses separate library in Jaguar and later

More Porting

- Add autoconf macros to test OS features
  - Lowest system version possible for shipping binaries
  - Read
    "GNU Autoconf, Automake and Libtool"
    (New Riders, 2000)
- AvailabilityMacros.h in Jaguar and later

Caveats

- Do not abuse root
  - Add tools to /usr/local/bin instead of overwriting
  - symlink headers instead of overwriting
- Mach-O object files very different
  - No standard GNU binutils port
- Watch out for whitespace in pathnames
- Watch out for HFS+ vs. UFS issues
  - foo.c, Foo.C, FoO.c in same directory
  - If INSTALL in dir, make install does nothing

Debugging With GDB

- Mac OS X debugger is GDB
- Currently based on FSF's 5.3 release
- Objective-C, Mach support, frameworks, etc.
- Many additional commands
  - future-break fmwk.c:432
  - print-object objc_obj
  - info mach-tasks

GCC

- Jaguar system compiler based on GCC 3.1
- 2.95.2 available for compatibility with 10.1
  - kext development
- Panther version of compiler based on GCC 3.3
- GCC 3.3 features
  - More optimization
  - Better C++ conformance
  - PCH for precompiled headers
  - Also available for Jaguar

Using GCC Versions

- 2.95.2 always at /usr/bin/gcc2
- Standard version 3 always at /usr/bin/gcc3
- /usr/bin/cc and /usr/bin/gcc symlink to one or other of gcc[23]
- Script gcc_select chooses
  - gcc_select 2
  - gcc_select 3
  - gcc_select 3.3
- Aux programs and headers also relinked

Using the Xcode IDE With Open Source Projects

Bodhi Gerfen
Technical Publications

What You'll Learn

- Two ways to use Xcode
- Benefits of each
- Steps for getting started

Two Types of Xcode Integration

- Build with GNU Makefile inside Xcode
- Use Xcode's native build system

Benefits of Makefile With Xcode

- Graphical IDE
- Symbol indexing
- Batch find
- Source control integration
- Documentation, man pages

Demo:
Xcode With a Makefile

Benefits of Xcode Native Build

- Graphical IDE
- Symbol indexing
- Batch find
- Source control integration
- Documentation, man pages
- Precompiled headers
- Integrated debugging
- Zero Link
- Predictive compilation
- Distributed builds
- Integrated performance tools
- CodeSense
- Fix and Continue

Demo:

Xcode's Native Build System

Wrap Up

Stan Shebs

To Sum Up

- Xcode is a complete tool environment
  - Familiar tools at the command line
  - More powerful tools at your disposal
- Standard porting techniques resolve API differences
- DarwinPorts can ease transition
- Take advantage of the Xcode IDE

For More Information

Reference Library

Documentation

UNIX Porting Guide
ADC Home > Documentation > Darwin & Core OS >

Packaging UNIX Software
ADC Home > Documentation > Darwin & Core OS >

GNU C/C++/Objective-C Compiler
ADC Home > Documentation > Tools >

GNU Preprocessor
ADC Home > Documentation > Tools

Reference Library (continued)

Documentation

Debugging with GDB
ADC Home > Documentation > Tools >

Assembler
ADC Home > Documentation > Tools

Macho-O Runtime Architecture
ADC Home > Documentation > Tools

Mac OS X Man Pages
ADC Home > Documentation > Darwin

Reference Library

Technical Notes

TN2071: “Porting Command Line UNIX Tools to Mac OS X”
ADC Home > Technical Notes >

Source Code

config.guess and config.sub source
http://savannah.gnu.org/cgi-bin/viewcvs/config/config/

Reference Library

Other Resources

ADC Tools pages
http://developer.apple.com/tools/

Darwin pages
http://www.apple.com/darwin/

DarwinPorts
http://opendarwin.org/projects/darwinports/

Fink at SourceForge
http://fink.sf.net/

Who to Contact

Godfrey DiGiorgi
Technology Manager, Development Tools
ramarren@apple.com

Development Tools Engineering Feedback
xcode-feedback@group.apple.com

Mailing Lists at Apple
http://lists.apple.com

http://developer.apple.com/wwdc2003/urls.html

Roadmap

102 Open Source at Apple Update on Apple's Open Source Infrastructure	**The Mission** **Tues., 10:30am**
603 WebObjects Technical Overview Understand the significance of WebObjects	**Russian Hill** **Tues., 10:30am**
302 Introduction to the Apple Dev Tools All about Xcode	**Presidio** **Tues., 3:30pm**
606 Mac OS X Server in Depth How to take advantage of Mac OS X Server	**The Marina** **Tues., 3:30pm**

Q&A

Godfrey DiGiorgi
Technology Manager, Development Tools
ramarren@apple.com

http://developer.apple.com/wwdc2003/urls.html

WWDC 2003

Worldwide Developers Conference